Project 1 - Variational Monte Carlo Methods for Bosonic Systems

Hannes Kneiding
FYS4411 - Computational Physics II

April 1, 2019

Link to GitHub - Repository

Abstract

A Variational Monte Carlo (VMC) approach has been developed to determine ground state energies of confined Bose systems with different numbers of particles and dimensions. Brute-force and importance sampling have been implemented. A non-interacting as well as an interacting system have been examined. Optimization of variational parameters has been done using the gradient descent method. The results have been evaluated using the statistical method of blocking. The one-body densities of particles have been determined and analyzed.
For the non-interacting system the analytical results could be reproduced using analytic expressions and numerical differentiation (with minor deviations). The obtained results (ground state energies) of the interacting system were in accordance with [1]. The usage of analytic expressions in favor of numerical differentiation proved to be crucial in order to save CPU time. The employment of importance sampling drastically increased acceptance rates and therefore efficiency of the simulation.

Contents

1 Introduction

The interest in studies of bosonic systems has increased since the demonstration of Bose-Einstein condensations (BEC) in gases of alkali atoms confined in magnetic traps. Research is aimed at obtaining a better understanding of the governing processes and the determination of defining physical quantities such as the critical temperature of BEC. [2]

A decisive characteristic of trapped alkali systems is that they are dilute. A system is considered to be dilute if the average distance between atoms is significantly larger than the range of interatomic interactions. The physics in dilute systems is dominated by two-body interactions, which are described by the s-wave scattering length a. Mathematically, the diluteness of a system can be defined by the the gas parameter $x(r) = n(r)\ a^3$, where $n(r)$ denotes the local density of the system. The Gross-Pitaevskii equation is applicable for average gas parameter values of $x_{av} \leq 10^3$. For higher values, more involved many-body approaches are required. [2]

In this project a Variational Monte Carlo (VMC) approach for the determination of the ground state energy of trapped BEC systems has been implemented. A simple non-interacting system as well as a more involved interacting system of bosons have been evaluated. The systems are characterized by trial wave functions with two variational parameters. The VMC has been implemented with brute-force Metropolis sampling and importance sampling. For the determination of the optimal variational parameters a gradient descent method has been chosen. The method of blocking has been used for the statistical analysis.

1.1 Problem definition

Our Hamiltonian is given by

$$\hat{H} = \sum_{i}^{N} \left(\frac{-\hbar^2}{2m} \nabla_i^2 + V_{ext}(\mathbf{r}_i) \right) + \sum_{i<j}^{N} V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{1}$$

with V_{ext} being the external potential

$$V_{ext}(\mathbf{r}) = \begin{cases} \frac{1}{2} m \omega_{ho}^2 r^2 & (S) \\ \frac{1}{2} m [\omega_{ho}^2 (x^2 + y^2) + \omega_z^2 z^2] & (E) \end{cases} \tag{2}$$

where S denotes the spherical and E the elliptic trap and ω_{ho}^2 being the potential strength; and V_{int} being the internal potential

$$V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \tag{3}$$

where a is the hard-core diameter of the bosons.

Our trial wave function for the ground state with N atoms is given by

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N, \alpha, \beta) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i) \right] \left[\prod_{j<k} f(a, |\mathbf{r}_j - \mathbf{r}_k|) \right], \tag{4}$$

where α and β are variational parameters. The single-particle wave function is proportional to the harmonic oscillator function for the ground state and is given by

$$g(\alpha, \beta, \mathbf{r}_i) = \exp\left[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right]. \tag{5}$$

The correlation wave function is given by

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a. \end{cases} \tag{6}$$

For a non-interacting system of bosons in a spherical harmonic oscillator we confine $a = 0$ and $\beta = 1$. For an interacting system of bosons in an elliptical potential we set $a \geq 0$ and use β as a variational parameter.

1.2 Benchmarks

For the non-interacting system, a multi-particle harmonic oscillator, analytic expressions can be derived which can be used to check the validity of the software. Assuming natural units the ground state energy of that system is

$$E_0 = \omega_{ho} \frac{dN}{2} \tag{7}$$

where N is the number of particles and d the number of spatial dimensions. Similarly the exact wave function equates to

$$\Psi = \prod_{i=1}^{N} \exp(-\frac{r_i^2}{2}) \tag{8}$$

Comparing this with the shape of the trial wave function (equation 5) it becomes apparent that setting the parameter $\alpha = 0.5$ will be the optimal choice during simulation.

For the interacting system there is no analytic solution available. Therefore, we benchmark our results with [1] who found

Table 1: Aprroximate benchmark values.

N	d	$\langle E_L \rangle$
10	3	24
50	3	130
100	3	270

On the other hand [3] found values of $E/N \approx 11.121\ \hbar\omega_{ho}$.

2 Methods

This project employs a Variational Monte Carlo (VMC) approach in order to determine the ground state energies of different bosonic systems. VMC simulations operate on the basis of the variational principle:

Given a trial wave function Ψ_T and a Hamiltonian $\hat{H}$, then the expectation value $\langle \hat{H} \rangle$ defined as

$$E[\hat{H}] = \langle \hat{H} \rangle = \frac{\int d\mathbf{R} \Psi_T^*(\mathbf{R}, \boldsymbol{\alpha}) \hat{H}(\mathbf{R}) \Psi_T(\mathbf{R}, \boldsymbol{\alpha})}{\int d\mathbf{R} \Psi_T^*(\mathbf{R}, \boldsymbol{\alpha}) \Psi_T(\mathbf{R}, \boldsymbol{\alpha})} \tag{9}$$

with variational parameters $\boldsymbol{\alpha}$, constitutes an upper bound to the exact ground state energy [4]:

$$E_0 \leq E[\hat{H}] \tag{10}$$

By varying the variational parameters $\boldsymbol{\alpha}$ and thereby obtaining various trial wave functions, approximations to the exact ground state energy can be made when analytical solutions to the ground state energy are not available.
However, the dimensionality of the integrals involved, directly scales with the number of particles and spatial dimensions, making the analytic calculation of the expectation value $\langle \hat{H} \rangle$ unfeasible for realistic systems.

2.1 Monte Carlo integration

Relief to this issue can be achieved by applying Monte Carlo integration for the evaluation of the integrals. Monte Carlo integration is based on the idea that an integral is nothing more than the area contained between the function itself and the axes or some kind of boundary. Therefore, an approximation to the exact integral can be obtained by randomly picking a series of points within the range of interest and then averaging over the corresponding function values. [5]
Let $f(\mathbf{x})$ be a function then the corresponding integral on the surface $\boldsymbol{\Omega}$ is defined as

$$I = \int_{\boldsymbol{\Omega}} f(\mathbf{x}) \, d\mathbf{x} \tag{11}$$

which can be approximated as

$$I \approx \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}_i) \tag{12}$$

where samples are uniformly drawn from $\boldsymbol{\Omega}$

$$\mathbf{x}_i \in \boldsymbol{\Omega} \tag{13}$$

The accuracy of the approximation obviously scales with the number of random points sampled (N). In the limit $N \to \infty$ the approximation becomes exact. In practice however, the number of samples needed to obtain a good approximation is unfeasibly high since all samples are generated from a uniform distribution and "important" areas may not get sampled enough.

2.2 Markov chains and Monte-Carlo Sampling

A more efficient approach is the employment of Markov-Chain-Monte-Carlo (MCMC) simulations for the sampling process. Markov chains are sequences of states, where each state $\mathbf{x}_i$ is only dependent on its previous state $\mathbf{x}_{i-1}$. In Monte Carlo simulations, Markov chains are generated by proposing a new state $\mathbf{x}'$ following some proposal rule and then deciding whether to accept or reject the trial state. [5] [6] If accepted, the proposed trial state $\mathbf{x}'$ is appended to the Markov chain; if rejected, the current state $\mathbf{x}_i$ will be appended to the Markov chain, thus building a chain of states. [5]

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ ... \ \mathbf{x}_{n-1}, \mathbf{x}_n\} \tag{14}$$

Each state is sampled from a probability distribution (PDF) $p(\mathbf{x}_{i+1}|\mathbf{x}_i)$ on the basis of the previous state. This PDF defines the transition probability from state $\mathbf{x}_i$) to state $\mathbf{x}_{i+1}$) and should be easy to sample from. MCMC thereby allows us to sample from a model instead of the original distribution from which direct sampling is difficult. [5, 6]

The basic MCMC process looks as follows:

1. Randomly initialize starting state x_0.
2. Repeat until convergence criterion is satisfied:
   (a) Propose trial state x' according to a specified proposal rule.
   (b) Accept or reject trial state according to a specified acceptance scheme.

2.2.1 Metropolis algorithm

By defining the local energy

$$E_L(\mathbf{R}) = \frac{1}{\Psi_T(\mathbf{R})}\hat{H}\Psi_T(\mathbf{R}), \tag{15}$$

and the PDF

$$P(\mathbf{R}) = \frac{|\Psi_T(\mathbf{R})|^2}{\int |\Psi_T(\mathbf{R})|^2 d\mathbf{R}} \tag{16}$$

the integral to be evaluated (equation 9) can be rewritten to

$$\langle E_L \rangle = \int P(\mathbf{R}) E_L(\mathbf{R}) d\mathbf{R} \tag{17}$$

This integral can now be approximated by Monte-Carlo integration

$$\langle E_L \rangle \approx \frac{1}{N} \sum_{i=1}^{N} P(\mathbf{R}_i) E_L(\mathbf{R}_i) \tag{18}$$

In order to obtain an expectation value for the local energy we need to sample states from P and accumulate the local energy values. For the MCMC sampling process an acceptance scheme is necessary. By splitting up the transition probability P into acceptance and proposal probability

$$p(\mathbf{x}_{i+1}|\mathbf{x}_i) = p_{acc}(\mathbf{x}_i \to \mathbf{x}_{i+1}) p_{prop}(\mathbf{x}_i \to \mathbf{x}_{i+1}) \tag{19}$$

the popular Metropolis acceptance scheme can be derived

$$p_{acc}(\mathbf{x}_i \to \mathbf{x}_j) = \min\left\{1,\ \frac{\mu(\mathbf{x}_j)p_{prop}(\mathbf{x}_j \to \mathbf{x}_i)}{\mu(\mathbf{x}_i)p_{prop}(\mathbf{x}_i \to \mathbf{x}_j)}\right\} \tag{20}$$

where $\mu(\mathbf{x}_i)$ is the probability of finding the system in state $\mathbf{x}_i$ and $p_{prop}(\mathbf{x}_i \to \mathbf{x}_j)$ is the probability of suggesting the specified trial position. [5, 6]
Assuming symmetric proposal probabilities

$$p_{prop}(\mathbf{x}_i \to \mathbf{x}_j) = p_{prop}(\mathbf{x}_j \to \mathbf{x}_i) \tag{21}$$

the acceptance scheme reduces to

$$p_{acc}(\mathbf{x}_i \to \mathbf{x}_j) = \min\left\{1,\ \frac{\mu(\mathbf{x}_j)}{\mu(\mathbf{x}_i)}\right\} \tag{22}$$

Inserting P into equation 22 yields the Metropolis acceptance scheme for this problem

$$p_{acc}(\mathbf{R}_i \to \mathbf{R}_j) = \min\left\{1,\ \frac{|\Psi_T(\mathbf{R_j})|^2}{|\Psi_T(\mathbf{R_i})|^2}\right\} \tag{23}$$

where the hard-to-compute integrals cancel out. As a proposal rule a uniform (and therefore symmetric) updating scheme has been chosen

$$\mathbf{R}' = \mathbf{R} + \gamma\mathbf{r} \tag{24}$$

where $\mathbf{r}$ is a vector of random numbers $\in [0, 1]$ according to the number of dimensions and γ is a chosen step size.

2.2.2 Importance sampling

The efficiency of Monte-Carlo sampling can be increased by replacing the brute-force Metropolis with a scheme where the random walk is biased by the trial wave function. [5] This approach is based on the Langevin and Fokker-Planck equations.
In one dimension the Langevin equation is given by

$$\frac{\partial x(t)}{\partial t} = DF(x(t)) + \eta \tag{25}$$

where D is the diffusion coefficient and η is a random variable. F denotes the drift vector defined as

$$F = 2\frac{1}{\Psi_T}\nabla\Psi_T \tag{26}$$

which can be derived from the Fokker-Planck equation. Integrating this differential equation with Euler's method yields an update equation for a trial position

$$y = x + DF(x)\Delta t + \xi\sqrt{\Delta t} \tag{27}$$

where Δt is a chosen time step and ξ is a Gaussian random variable. The drift vector pushes the trial particles to regions where the wave function is large so that the proposal probabilities become asymmetric.

Therefore, they have to be considered in the Metropolis acceptance scheme. The solution to the Fokker-Planck equation yields the required proposal probabilities expressed in terms of Green's function

$$G(y, x, \Delta t) = \frac{1}{(4\pi D\Delta t)^{3N/2}} \exp(-(y - x - D\Delta t F(x))^2 / 4D\Delta t) \tag{28}$$

Therefore, the Metropolis choice is replaced by the Metropolis-Hastings article

$$p_{acc}(\mathbf{R}_i \to \mathbf{R}_j) = \min\left\{1,\ \frac{G(\mathbf{R}_i, \mathbf{R}_j, \Delta t)|\Psi_T(\mathbf{R}_j)|^2}{G(\mathbf{R}_j, \mathbf{R}_i, \Delta t)|\Psi_T(\mathbf{R}_i)|^2}\right\} \tag{29}$$

Cancelling out the prefactors in Green's function and merging the exponentials yields a simplified and less expensive to compute expression

$$p_{acc}(\mathbf{R}_i \to \mathbf{R}_j) = \min\left\{1,\ \Lambda\frac{|\Psi_T(\mathbf{R}_j)|^2}{|\Psi_T(\mathbf{R}_i)|^2}\right\} \tag{30}$$

with

$$\Lambda = \exp\left(0.5(\mathbf{R}_i - \mathbf{R}_j)(\mathbf{F}(\mathbf{R}_i) + \mathbf{F}(\mathbf{R}_j)) + \frac{D\Delta t}{4}([\mathbf{F}(\mathbf{R}_i)]^2 - [\mathbf{F}(\mathbf{R}_j)]^2)\right) \tag{31}$$

2.3 Gradient descent

A crucial step in simulating a bosonic system with a VMC approach is finding the optimal values for all variational parameters $\boldsymbol{\alpha}$. In principle this can be done by employing a brute-force search over the possible values of $\boldsymbol{\alpha}$ in order to find the best set of parameters. This approach however, is extremely inefficient and time consuming. Instead a more intelligent scheme known as gradient descent can be used to solve this optimization problem. The gradient descent algorithm uses the gradient in order to iteratively update the parameters. [7] It is defined by the update step

$$\boldsymbol{\alpha}_{i+1} = \boldsymbol{\alpha}_i - \gamma \nabla_\alpha \mathbf{F}(\boldsymbol{\alpha}_i) \tag{32}$$

where $\mathbf{F}$ is the function to be optimized and γ is the step size, which scales the gradient and determines the speed of adaptation. [7] The gradient can be updated during the optimization to avoid overshooting and slow convergence. The optimization starts from some randomly initialized point $\boldsymbol{\alpha}_0$ and ends when a convergence criterion is satisfied (typically gradient is below a certain threshold). [7]
In this case the function to optimize is the expectation value of the local energy $\langle E_L(\boldsymbol{\alpha})\rangle$. To determine its gradient with respect to $\boldsymbol{\alpha}$ define

$$\bar{E}_{\boldsymbol{\alpha}} = \frac{d\langle E_L(\boldsymbol{\alpha})\rangle}{d\boldsymbol{\alpha}} \tag{33}$$

and

$$\bar{\Psi}_T = \frac{d\Psi_T(\boldsymbol{\alpha})}{d\boldsymbol{\alpha}} \tag{34}$$

By applying the chain rule and using the hermicity of the Hamiltonian it can be shown that

$$\bar{E}_{\boldsymbol{\alpha}} = 2\left(\left\langle \frac{\bar{\Psi}_T}{\Psi_T(\boldsymbol{\alpha})} E_L(\boldsymbol{\alpha})\right\rangle - \left\langle \frac{\bar{\Psi}_T}{\Psi_T(\boldsymbol{\alpha})}\right\rangle \langle E_L(\boldsymbol{\alpha})\rangle\right) \tag{35}$$

During the optimization we therefore need to keep track of

$$\frac{\bar{\Psi}_T}{\Psi_T(\boldsymbol{\alpha})} E_L(\boldsymbol{\alpha}) \tag{36}$$

and

$$\frac{\bar{\Psi}_T}{\Psi_T(\boldsymbol{\alpha})} \tag{37}$$

to be able to compute the expectation values needed for the derivative. Every $10^4 - 10^5$ Monte Carlo cycles a new gradient step is computed and the parameter $\boldsymbol{\alpha}$ is updated.

2.4 Blocking

Monte Carlo methods typically produce long sequences of data points, where each data point can be interpreted as an experimental measurement. The obtained data points should be statistically analyzed and evaluated to determine the expectation value of the quantity of interest and its estimated error margin, which can be calculated from the variance. [2]
The variance for an uncorrelated sample

$$var(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 \tag{38}$$

can be easily computed, but gives a too optimistic approximation of the true variance and therefore the error. The reason for this is that Monte Carlo simulations produce correlated samples because the individual steps are linked together in a Markov chain. [2]
For correlated samples the sample error is defined as

$$err_x^2 = \frac{1}{N} cov(x) \tag{39}$$

with the covariance

$$cov(x) = \frac{1}{N} \sum_{i,j}^{N} (x_i - \mu)(x_j - \mu) \tag{40}$$

Rewriting this expression using the autocorrelation function κ_d yields

$$err_x^2 = \frac{\tau}{N}\ var(x) \tag{41}$$

where τ is the autocorrelation time

$$\tau = 1 + 2 \sum_{d=1}^{N-1} \kappa_d \tag{42}$$

with

$$\kappa_d = \frac{f_d}{var(x)} \tag{43}$$

and

$$f_d = \frac{1}{n-d} \sum_{k=1}^{n-d} (x_k - \mu)(x_{k-d} - \mu) \tag{44}$$

The computation of the autocorrelation time is very expensive since it contains the evaluation of a double sum over all particles and therefore has a complexity of $\mathcal{O}(N^2)$.
The calculation of this quantity can be circumvented by iteratively applying a so-called blocking transformation proposed in [8], that repeatedly halves the length of the data set. [8] The transformation is given as

$$x_i' = \frac{1}{2}(x_{2i-1} + x_{2i}) \tag{45}$$

so that

$$n' = \frac{1}{2}n \tag{46}$$

The sample error err_x^2 is invariant under this operation, which leads to the inequality [8]

$$err_x^2 \geq \left\langle \frac{f_0}{n-1} \right\rangle \tag{47}$$

It can be shown that there exists a fixed point for which the equality holds. [8]
Equation (47) is evaluated for decreasing block size according to the blocking transformation by approximating the expectation value with the fraction $\frac{f_0}{n-1}$ itself. This process is repeated until the block size is $n' = 2$. (Because the block size is halved in each iteration it is advisable to have data sets with lengths that are powers of 2.) The values for $\frac{f_0}{n-1}$ will increase with decreasing block size. As soon as a fixed point is reached the values will converge and form a plateau with constant values within an error margin. The highest value of this plateau is the estimation of the sample error err_x^2. [8]

In this project an automated approach to the blocking method proposed in [9] is used.

2.5 One-body density

The one-body density is defined as

$$\rho(\mathbf{r}_1) = \int d\mathbf{r}_2 \dots d\mathbf{r}_N |\Psi(\mathbf{r}_1 \dots \mathbf{r}_N)|^2 \tag{48}$$

and is a measure for the distribution of particles in a given system. Therefore, the product $\rho(\mathbf{r}_1)d\mathbf{r}_1$ is the probability of finding any particle within the volume $d\mathbf{r}_1$ [10]
In VMC calculations the one-body density can be obtained by saving snapshots of the system during the simulation and then using the coordinates to produce a histogram.

$$\rho(\mathbf{r}_1) = \frac{H(\mathbf{r}_1)}{r_1^{(d-1)}} \tag{49}$$

where d is the number of dimensions. [10]

3 Implementation

The VMC algorithm has been implemented in *Python* using an object-oriented approach. For random number generation and vector operations the package *numpy* is used. In the following all classes and their functionalities are briefly outlined:

- **Wave Function** - Static class that contains analytic (and numerical) expressions for gradients, Laplacians and derivatives of the respective wave function.
  - **Gaussian**
  - **InteractingGaussian**
- **Energy Model** - Class that represents the chosen Hamiltonian by defining expressions for the kinetic, potential and (if present) interaction energies. Needs to be initialized with a specified wave function.
  - **HarmonicOscillator**
  - **InteractingHarmonicOscillator**
- **Particle** - Custom data type for a particle containing its position and methods to propose a trial position.
- **System** - Custom data type for a system of particles containing a list of all particles and functions to evaluate the wave function, local energy and drift force. Needs to be initialized with a specified energy model.
- **Simulation** - Main class for running a VMC simulation. Contains all major, high level functionalities (MC sampler, gradient descent method). Needs to be initialized with a specified system.
- **Observables** - Custom data type for handling energy expectation values during the simulation.

Furthermore, two independent scripts for the calculation of the error using the method of blocking[1] and for the computation of the one-body density are included in a separate directory.

[1] This script was provided by the supervisor [2].

4 Results and Discussion

This chapter is divided into two parts for the analysis of the non-interacting and interacting system respectively.

4.1 Non-interacting System

The non-interacting system was analyzed for particle numbers of $N = 1$, $N = 10$, $N = 100$ and $N = 500$ in one, two and three dimensions. The performances of analytical and numerical derivation for the computation of the Laplacian of the wave function were compared. Furthermore, the effect of importance sampling on the results and the acceptance rate was investigated. The analysis of the non-interacting system also serves as a measure to verify the correctness of the implementations.
A brute-force search over the possible values of the variational parameter α gives an optimal value of 0.5, which was expected from the analytical solution of the non-interacting harmonic oscillator (equation 8). For the calculations atomic units ($m = 1$) have been assumed. The chosen metropolis step with was 1.5.

Table 2 and 3 show the obtained results for the analytical and numerical approach respectively.

Table 2: Results of the analytical approach using brute-force sampling.

N	d	$\langle E_L \rangle$	σ	CPU time (s)
1	1	0.5	0	0.140
1	2	1	0	0.156
1	3	1.5	0	0.172
10	1	5	0	1.326
10	2	10	0	1.622
10	3	15	0	1.919
100	1	50	0	63.804
100	2	100	0	83.071
100	3	150	0	98.499
500	1	250	0	1449.499
500	2	500	0	1817.490
500	3	750	0	2228.973

Table 3: Results of the numerical approach using brute-force sampling.

N	d	$\langle E_L \rangle$	σ	CPU time (s)
1	1	0.5	$3.736 \cdot 10^{-9}$	0.156
1	2	1	$4.794 \cdot 10^{-9}$	0.203
1	3	1.5	$5.205 \cdot 10^{-9}$	0.281
10	1	5	$6.529 \cdot 10^{-9}$	2.215
10	2	10	$1.306 \cdot 10^{-8}$	3.822
10	3	15	$1.686 \cdot 10^{-8}$	5.897
100	1	50	$3.016 \cdot 10^{-8}$	133.615
100	2	100	$4.265 \cdot 10^{-8}$	260.350
100	3	150	$2.000 \cdot 10^{-7}$	430.828
500	1	250	$8.530 \cdot 10^{-8}$	3138.725
500	2	500	$4.514 \cdot 10^{-7}$	6082.744
500	3	750	$7.629 \cdot 10^{-7}$	11838.276

The simulation with analytic expressions for the Laplacian produces exact ground state energies with no error at all. The numerical calculations give similar results only with minor deviations. These are due to the finite accuracy of numerical differentiation. The error increases with increasing number of particles and dimensions.
Intuitively, the required computational time increases with increasing number of particles and dimensions as well.

Table 4 and 5 show the obtained results for the analytical and numerical approach when using importance sampling.

Table 4: Results of the analytical approach using importance sampling.

N	d	$\langle E_L \rangle$	σ	CPU time (s)
1	1	0.5	0	0.203
1	2	1	0	0.203
1	3	1.5	0	0.234
10	1	5	0	0.998
10	2	10	0	1.139
10	3	15	0	1.295
100	1	50	0	34.913
100	2	100	0	42.448
100	3	150	0	51.153
500	1	250	0	761.737
500	2	500	0	948.268
500	3	750	0	1097.888

Table 5: Results of the numerical approach using importance sampling.

N	d	$\langle E_L \rangle$	σ	CPU time (s)
1	1	0.5	$1.129 \cdot 10^{-9}$	0.218
1	2	1	$1.917 \cdot 10^{-9}$	0.265
1	3	1.5	$1.563 \cdot 10^{-9}$	0.328
10	1	5	$1.686 \cdot 10^{-9}$	1.810
10	2	10	$2.384 \cdot 10^{-9}$	3.292
10	3	15	$4.768 \cdot 10^{-9}$	5.086
100	1	50	$3.372 \cdot 10^{-8}$	100.293
100	2	100	$5.223 \cdot 10^{-8}$	220.398
100	3	150	$7.864 \cdot 10^{-8}$	388.474
500	1	250	$1.594 \cdot 10^{-7}$	2339.516
500	2	500	$1.706 \cdot 10^{-7}$	5215.316
500	3	750	$4.825 \cdot 10^{-7}$	9975.078

The obtained results for both approaches are very similar to the results obtained from brute-force sampling. The main difference is the required CPU time, which is larger compared to brute-force sampling due to the additional computational overhead from evaluating the drift vectors and Green's function. Furthermore, the error in the numerical approach is slightly lower in importance sampling.

Table 6 and 7 show the average acceptance rates for different time steps in importance sampling for the analytical and numerical approach respectively.

Table 6: Results of the analytical approach using different time steps.

Δt	$\langle E_L \rangle$	σ	Acceptance rate
0.001	150	0	0.9999
0.005	150	0	0.9998
0.01	150	0	0.9996

Table 7: Results of the numerical approach using different time steps.

Δt	$\langle E_L \rangle$	σ	Acceptance rate
0.001	150	$2.257 \cdot 10^{-7}$	0.9999
0.005	150	$2.955 \cdot 10^{-7}$	0.9998
0.01	150	$1.809 \cdot 10^{-7}$	0.9995

In both cases the energies are not affected by the chosen time step. In the numerical case the error slightly decreases with increasing time steps.
The acceptance rates are in both cases extremely high almost reaching 100%. Compared to the average acceptance rates of 60 to 80% obtained in brute-force sampling this is a substantial increase.

Repeating the error estimation for the obtained results with the method of blocking only yielded values marginally different from the values obtained with the empirical approach. Therefore, these results are not include here. For the interacting system however, solely the method of blocking has been used for error estimation.

4.2 Interacting System

Fixing $\beta = 2.82843$ as well as $a/a_{ho} = 0.0043$[2] and conducting a brute-force search over values of the variational parameter α showed that the optimal value lies somewhere between 0.49 and 0.5. A gradient descent optimization yielded a value within that range, $\alpha = 0.49731$. The found value for α and the parameterization $\gamma = \beta = 2.82843$ have been used for the subsequent production runs of the VMC. The chosen metropolis step width was 1.5.

Table 8 shows the results obtained for the three-dimensional system with different particle numbers using the full interacting Hamiltonian and wave function. The sample error σ has been determined using the blocking method.

Table 8: Results of the interacting system.

N	d	$\langle E_L \rangle$	σ
10	3	24.399	$3.716 \cdot 10^{-4}$
50	3	127.281	0.1
100	3	271.306	1.609

From the results it can be observed that the error increases with increasing number of particles. Furthermore, the energy values show significant differences from the energies obtained for the non-interacting system. They are about 50% higher than the values of the non-interacting system. This energy contribution hails from the interacting, repulsive potential. The energy values of the interacting system also show substantially higher sample errors compared to the energies of the non-interacting system.

The results show significant differences to the results of [3] which is most likely due to different parameterization (different value for the hard-core diameter of bosons a) and different choice of variational parameters (α). At the same time the obtained results match the results of [1] almost precisely with minor deviations.

4.3 Comparison of Systems

Figure 1 and 2 show the calculated one-body densities for a particle number of $N = 10$ of the non-interacting and interacting system respectively.

When comparing the two one-body density plots it becomes apparent that they both have very similar shape. Their difference however lies in the distribution centers. The non-interacting system has an average difference at around 1.1 and its distribution is also slightly shifted to the right compared with the interacting system. The interacting system has an average distance of 1. The difference of the distributions hails from the usage of different traps. The spherical trap allows particles to arrange within a sphere of specified radius, whereas the elliptic trap forces the particles to arrange in an elliptic form. This of course influences the overall particle distribution which finds expression in the one-body densities.

[2] The derivation of the simplified Hamiltonian can be found in chapter 6.

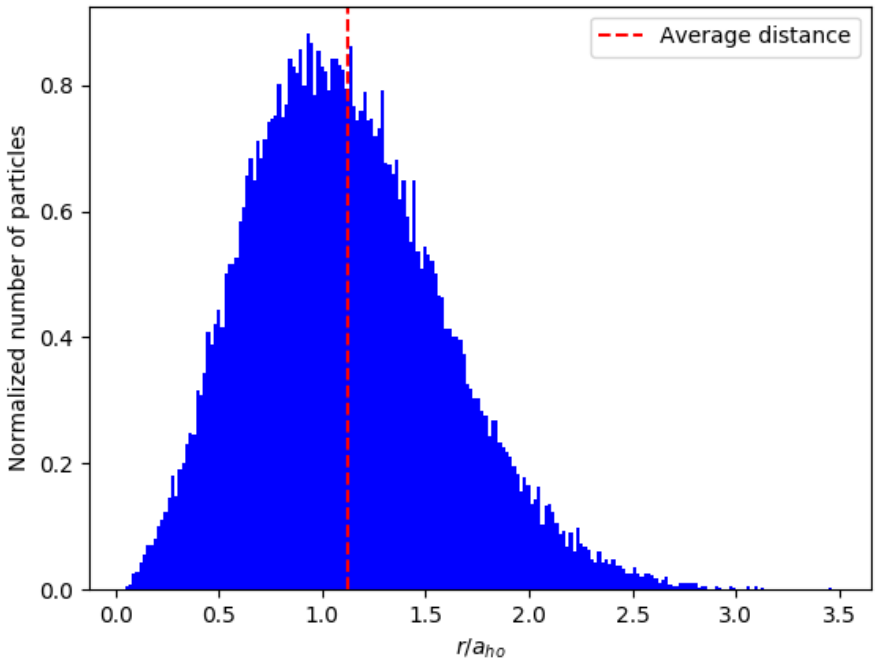

Figure 1: One-body density of the non-interacting system.

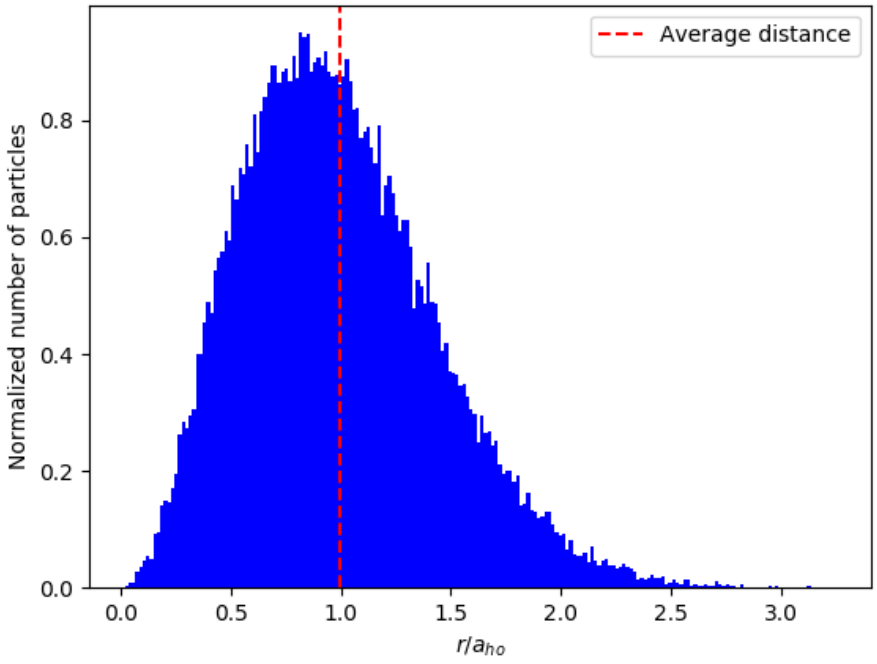

Figure 2: One-body density of the interacting system.

5 Conclusion

The analysis of the non-interacting system showed that the VMC algorithm sampled correctly and produced the exact analytical results. Optimization of the variational parameter alpha yielded the same result as obtained from the analytical solution to the harmonic oscillator. The implementation of importance sampling substantially increased the sampling efficiency by reducing the amount of rejected steps. Furthermore, it could be shown that the use of numerical differentiation for the computation of the Laplacian drastically increases the required CPU time. The use of analytical expressions for the calculation of derivatives is therefore always favorable.

The obtained results of the interacting system were in accordance with [1]. Due to the computational complexity of the computation of the Laplacian ($\mathcal{O}(N^3)$) the required CPU times for the evaluation of the interacting system were significantly higher compared to the non-interacting case. The one-body densities of the interacting and non-interacting systems showed small but considerable differences in their distributions which were expected and can be attributed to the additional *Jastrow* factor in the wave function of the interacting system. In this setting the influence of the *Jastrow* factor is marginal because of the small number of particles in the system. If the particle density within the system increases the interbosonic interactions become more important and have greater impact on the final results.

The programming language *python* proved to be a simple and flexible means for the implementation of a scientific application like this VMC algorithm. The possibility for object-oriented programming enables the creation of structured and maintainable code. However, when it comes to performance and required CPU time *python* cannot quite compete with low level programming languages such as *C++* that excel at executing loops and simple mathematical operations. This problem can be counteracted by using *python* libraries such as *numba* that allows just-in-time compilation of *python* code into fast machine code.

6 Derivation of analytic Expressions

A great part of the simulation of quantum systems using a VMC approach evolves around the computation of the local Energy:

$$E_L(\mathbf{R}) = \frac{1}{\Psi_T(\mathbf{R})}\hat{H}\Psi_T(\mathbf{R}), \tag{50}$$

with a specified trial wave function $\Psi_T(\mathbf{R})$ and Hamiltonian $\hat{H}$. The computation of the local energy normally requires the calculation of the second derivative of the wave function, which can be either determined analytically or numerically. If possible an analytical approach is always favorable since numerical differentiation is slow and inaccurate. In the following, the derivations of the analytical expressions of the local energies are outlined. Furthermore, derivations of the expressions of the drift force, derivatives with respect to the variational parameters and the simplified Hamiltonian are discussed.

6.1 Local Energy for non-interacting Systems

First the local energy for the harmonic oscillator ($a = 0$, $\beta = 1$) for one particle in the one dimensional setting is calculated. For simplicity natural units are assumed (i.e. $\hbar = m = 1$).

One dimensional case
The trial wave function and the corresponding Hamiltonian for the one dimensional case are

$$\Psi_T(x) = e^{-\alpha x^2} \tag{51}$$

and

$$\hat{H} = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + \frac{1}{2}\omega_{ho}x^2 \tag{52}$$

The second derivative of the trial wave function is

$$\frac{\partial^2 \Psi_T}{\partial x^2} = 2\alpha e^{-\alpha x^2}(2\alpha x^2 - 1) \tag{53}$$

Therefore, the complete expression for the local energy is

$$E_L = \frac{1}{\Psi_T}\hat{H}\Psi_T = \alpha(1 - 2\alpha x^2) + \frac{1}{2}\omega_{ho}^2 x^2 \tag{54}$$

Two dimensional case
The trial wave function and the corresponding Hamiltonian for the two dimensional case are

$$\Psi_T(x, y) = e^{-\alpha(x^2+y^2)} \tag{55}$$

and

$$\hat{H} = -\frac{1}{2}(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}) + \frac{1}{2}\omega_{ho}(x^2 + y^2) \tag{56}$$

The second (partial) derivative of the trial wave function is

$$(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})\Psi_T = 2\alpha e^{-\alpha(x^2+y^2)}(2\alpha(x^2 + y^2) - 2) \tag{57}$$

Therefore, the complete expression for the local energy is

$$E_L = \frac{1}{\Psi_T}\hat{H}\Psi_T = \alpha(2 - 2\alpha x^2) + \frac{1}{2}\omega_{ho}^2(x^2 + y^2) \tag{58}$$

Three dimensional case
The trial wave function and the corresponding Hamiltonian for the three dimensional case are

$$\Psi_T(x, y, z) = e^{-\alpha(x^2+y^2+z^2)} \tag{59}$$

and

$$\hat{H} = -\frac{1}{2}(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}) + \frac{1}{2}\omega_{ho}(x^2 + y^2 + z^2) \tag{60}$$

The second (partial) derivative of the trial wave function is

$$(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2})\Psi_T = 2\alpha e^{-\alpha(x^2+y^2+z^2)}(2\alpha(x^2 + y^2 + z^2) - 3) \tag{61}$$

Therefore, the complete expression for the local energy is

$$E_L = \frac{1}{\Psi_T}\hat{H}\Psi_T = \alpha(3 - 2\alpha x^2) + \frac{1}{2}\omega_{ho}^2(x^2 + y^2 + z^2) = \alpha(3 - 2\alpha x^2) + \frac{1}{2}\omega_{ho}^2(\mathbf{r}^2) \tag{62}$$

6.1.1 Many Particle Setting

In a many particle setting the wave function and Hamiltonian would be

$$\Psi_T(\mathbf{R}) = \prod_i e^{-\alpha \mathbf{r}_i^2} \tag{63}$$

$$\hat{H} = \sum_i^N \left(-\frac{1}{2}\bigtriangledown_i^2 + \frac{1}{2}\omega^2 \mathbf{r}_i^2\right) \tag{64}$$

where $\mathbf{R}$ is the collection of particle positions $\mathbf{r}_i$. The first derivative of the trial wave function with respect to particle k is

$$\bigtriangledown_k \prod_i e^{-\alpha \mathbf{r}_i^2} = -2\alpha \mathbf{r}_k \prod_i e^{-\alpha \mathbf{r}_i^2} \tag{65}$$

Therefore, the second derivative with respect to particle k evaluates to

$$\bigtriangledown_k^2 \prod_i e^{-\alpha \mathbf{r}_i^2} = 2\alpha(2\alpha \mathbf{r}_k^2 - 3) \prod_i e^{-\alpha \mathbf{r}_i^2} \tag{66}$$

The resulting full expressions for the local energy in one, two and three dimensions then are

$$E_L = \sum_i^N \left(\alpha(1 - 2\alpha x^2) + \frac{1}{2}\omega_{ho}^2 x^2\right) \tag{67}$$

$$E_L = \sum_i^N \left(\alpha(2 - 2\alpha r_i^2) + \frac{1}{2}\omega_{ho}^2 r_i^2\right) \tag{68}$$

$$E_L = \sum_i^N \left(\alpha(3 - 2\alpha r_i^2) + \frac{1}{2}\omega_{ho}^2 r_i^2\right) \tag{69}$$

6.2 Local Energy for interacting Systems

The trial wave function for interacting systems is

$$\Psi_T(\alpha,\beta,\mathbf{R}) = \prod_i g(\alpha,\beta,\mathbf{r}_i)\exp\left(\sum_{i<j} u(r_{ij})\right) \tag{70}$$

with $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ and $u(r_{ij}) = \ln f(r_{ij})$. Furthermore we set

$$g(\alpha,\beta,\mathbf{r}_i) = \exp\big(-\alpha(x_i^2 + y_i^2 + \beta z_i^2\big) = \phi(\mathbf{r}_i) \tag{71}$$

The trial wave function can then be simplyfied to

$$\Psi_T(\mathbf{R}) = \prod_i \phi(\mathbf{r}_i)\exp\left(\sum_{i<j} u(r_{ij})\right) \tag{72}$$

To obtain the second derivative of the wave function we first calculate the first derivative, the gradient, of the wave function. Therefore, we calculate the individual derivatives of the two parts of the wave function. These are

$$\nabla_k \prod_i \phi(\mathbf{r}_i) = \nabla_k\phi(\mathbf{r}_k)\prod_{i\neq k}\phi(\mathbf{r}_i) \tag{73}$$

for the product and

$$\nabla_k \exp\left(\sum_{i<j} u(r_{ij})\right) = \exp\left(\sum_{i<j} u(r_{ij})\right)\nabla_k \sum_{i<j} u(r_{ij}) \tag{74}$$

$$= \exp\left(\sum_{i<j} u(r_{ij})\right)\sum_{j\neq k}\nabla_k u(r_{kj}) \tag{75}$$

for the exponential, with

$$\nabla_k u(r_{kj}) = \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}}\frac{\partial u(r_{kj})}{\partial r_{kj}} = \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}}u'(r_{kj}) \tag{76}$$

which follows from the chain rule and the fact that r_{kj} is the Euclidean distance between particle k and j. Using these building blocks and applying the product rule yields the expression for the gradient of the wave function

$$\nabla_k\Psi_T(\mathbf{R}) = \nabla_k\phi(\mathbf{r}_k)\left[\prod_{i\neq k}\phi(\mathbf{r}_i)\right]\exp\left(\sum_{i<j}u(r_{ij})\right) + \prod_i\phi(\mathbf{r}_i)\exp\left(\sum_{i<j}u(r_{ij})\right)\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{kj}) \tag{77}$$

Differentiating this expression once more will yield the second derivative of the wave function in question. Differentiating the first term gives

$$\prod_{i\neq k}\phi(\mathbf{r}_i)\exp\left(\sum_{i<j}u(r_{ij})\right)\left[\nabla_k^2\phi(\mathbf{r}_k) + \nabla_k\phi(\mathbf{r}_k)\left(\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{kj})\right)\right] \tag{78}$$

Dividing by Ψ_T gives

$$\frac{\nabla_k^2\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + \frac{\nabla_k\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)}\left(\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{kj})\right) \tag{79}$$

Next, the second term is differentiated

$$\prod_{i\neq k}\phi(\mathbf{r}_i)\Bigg\{\nabla_k\phi(\mathbf{r}_k)\exp\left(\sum_{i<j}u(r_{ij})\right)\sum_{j\neq k}\nabla_k u(r_{kj}) \quad + \tag{80}$$

$$\phi(\mathbf{r}_k)\exp\left(\sum_{i<j}u(r_{ij})\right)\left[\sum_{i\neq k}\nabla_k u(r_{ki})\sum_{j\neq k}\nabla_k u(r_{kj}) + \sum_{j\neq k}\nabla_k^2 u(r_{kj})\right]\Bigg\} \tag{81}$$

Dividing by Ψ_T gives

$$\frac{\nabla_k\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)}\sum_{j\neq k}\nabla_k u(r_{kj}) + \left[\sum_{i\neq k}\nabla_k u(r_{ki})\sum_{j\neq k}\nabla_k u(r_{kj}) + \sum_{j\neq k}\nabla_k^2 u(r_{kj})\right] \tag{82}$$

The very first term equates to

$$\frac{\nabla_k\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)}\left(\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{kj})\right) \tag{83}$$

which can be merged with the second term of 79. The first term in the square brackets evaluates to (compare to 76)

$$\sum_{i,j\neq k}(\frac{(\mathbf{r}_k-\mathbf{r}_i)(\mathbf{r}_k-\mathbf{r}_j)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) \tag{84}$$

Since

$$\nabla_k^2 u(r_{kj}) = \nabla_k\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{kj}) = \nabla_k\left[\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}\right]u'(r_{kj}) + \frac{(\mathbf{r}_k-\mathbf{r}_j)^2}{r_{kj}^2}u''(r_{kj}) \tag{85}$$

$$= \frac{3r_{kj}^2-(\mathbf{r}_k-\mathbf{r}_j)^2}{r_{kj}^3}u'(r_{kj}) + \frac{(\mathbf{r}_k-\mathbf{r}_j)^2}{r_{kj}^2}u''(r_{kj}) \tag{86}$$

$$= \frac{(\mathbf{r}_k-\mathbf{r}_j)(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}^2}\left(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right) \tag{87}$$

$$= u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj}) \tag{88}$$

the second term in the square brackets evaluates to

$$\sum_{j\neq k}\left(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right) \tag{89}$$

again using the fact that r_{kj} is the Euclidean distance between particle k and j.

Putting these results together (79, 83, 84, 89), the complete expression for the second derivative of the wave function divided by the wave function is

$$\frac{1}{\Psi_T(\mathbf{R})}\nabla_k^2\Psi_T(\mathbf{R}) = \frac{\nabla_k^2\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2\frac{\nabla_k\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)}\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{kj}) \quad + \tag{90}$$

$$\sum_{i,j\neq k}(\frac{(\mathbf{r}_k-\mathbf{r}_i)(\mathbf{r}_k-\mathbf{r}_j)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) + \sum_{j\neq k}\left(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right) \tag{91}$$

6.3 Drift Force

We furthermore calculate the analytic expression of the drift force F which is used in importance sampling for both the non-interacting and interacting system.

$$F = \frac{2\nabla\Psi_T}{\Psi_T} \tag{92}$$

6.3.1 Non-interacting System

The trial wave function for the non-interacting system in three dimensions is

$$\Psi_T(x,y,z) = e^{-\alpha(x^2+y^2+z^2)} \tag{93}$$

The first (partial) derivative of the trial wave function is

$$(\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z})\Psi_T = -2\alpha e^{-\alpha(x^2+y^2+z^2)}(x,y,z) \tag{94}$$

The drift force then is

$$F = -4\alpha(x+y+z) = -4\alpha\mathbf{r} \tag{95}$$

The same expression also holds for the one- and two-dimensional case where $\mathbf{r}$ then is a scalar or two-dimensional vector.

6.3.2 Interacting System

The drift force for the interacting system can be easily determined using the expression for the gradient of the wave function computed earlier (equation 77). Dividing the gradient by the wave function itself and multiplying with two yields the drift force for the interacting system.

$$\mathbf{F} = 2\left(\frac{\nabla_k\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + \sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{kj})\right) \tag{96}$$

6.4 Derivatives with respect to the variational parameters

In order to be able to use the gradient descent method for the optimization step, the derivatives of the wave functions with respect to the variational parameters must be determined.

6.4.1 Non-interacting System

For the non-interacting system only the variational parameter α is considered. The wave function for the non-interacting system is given by

$$\Psi_T(\mathbf{R}) = \prod_i \exp\left(-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right) \tag{97}$$

Therefore, the derivative with respect to α is

$$\frac{\partial}{\partial\alpha}\Psi_T(\mathbf{R}) = -\left[\sum_i (x_i^2 + y_i^2 + \beta z_i^2)\right]\left[\prod_i \exp\left(-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right)\right] \tag{98}$$

6.4.2 Interacting System

For the interacting system both variational parameters α and β are considered. The wave function for the interacting system is given by

$$\Psi_T(\mathbf{R}) = \left[\prod_i \exp\left(-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right)\right]\left[\exp\left(\sum_{i<j} u(r_{ij})\right)\right] \tag{99}$$

The second exponential does not depend on any of the two variational parameters α or β and can therefore be treated as a factor.
The derivative with respect to α is

$$\frac{\partial}{\partial\alpha}\Psi_T(\mathbf{R}) = -\left[\sum_i (x_i^2 + y_i^2 + \beta z_i^2)\right]\left[\prod_i \exp\left(-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right)\right]\left[\exp\left(\sum_{i<j} u(r_{ij})\right)\right] \tag{100}$$

The derivative with respect to β is

$$\frac{\partial}{\partial\alpha}\Psi_T(\mathbf{R}) = -\left[\sum_i (z_i^2)\right]\left[\prod_i \exp\left(-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right)\right]\left[\exp\left(\sum_{i<j} u(r_{ij})\right)\right] \tag{101}$$

6.5 Derivation of the simplified Hamiltonian

The full Hamiltonian for the interacting system is given by

$$\hat{H} = \sum_i^N \left(\frac{-\hbar^2}{2m}\nabla_i^2 + \frac{1}{2}m\left[\omega_{ho}^2(x_i^2 + y_i^2) + \omega_z^2 z_i^2\right]\right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{102}$$

Introducing lengths in terms of a_{ho} such that $\tilde{r} = r/a_{ho}$ with

$$a_{ho} = \sqrt{\frac{\hbar}{m\omega_{ho}}} \tag{103}$$

yields

$$\hat{H} = \sum_i^N \frac{1}{2}\left(\frac{-\hbar^2}{a_{ho}^2 m}\tilde{\nabla}_i^2 + a_{ho}^2 m\left[\omega_{ho}^2(\tilde{x}_i^2 + \tilde{y}_i^2) + \omega_z^2\tilde{z}_i^2\right]\right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{104}$$

where

$$\nabla_i^2 = \frac{1}{a_{ho}^2}\widetilde{\nabla}_i^2 = \frac{1}{a_{ho}^2}\left(\frac{\partial^2}{\partial(\frac{x_i}{a_{ho}})^2} + \frac{\partial^2}{\partial(\frac{y_i}{a_{ho}})^2} + \frac{\partial^2}{\partial(\frac{z_i}{a_{ho}})^2}\right) = \frac{1}{a_{ho}^2}\left(\frac{\partial^2}{\partial\tilde{x}_i^2} + \frac{\partial^2}{\partial\tilde{x}_i^2} + \frac{\partial^2}{\partial\tilde{x}_i^2}\right) \tag{105}$$

due to the chain rule.
The repulsive potential V_{int} stays unaffected since the hard-core diameter of the bosons a is rescaled as well as the positions to $a = a/a_{ho}$.
Inserting (103) into the Hamiltonian and assuming atomic units ($m = 1$) yields

$$\hat{H} = \sum_i^N \frac{1}{2}\left(-\hbar\omega_{ho}\widetilde{\nabla}_i^2 + \hbar\Big[\omega_{ho}(\tilde{x}_i^2 + \tilde{y}_i^2) + \frac{\omega_z^2}{\omega_{ho}}\tilde{z}_i^2\Big]\right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{106}$$

In order to express energy in units of $\hbar\omega_{ho}$ the whole expression is divided by $\hbar\omega_{ho}$ leading to

$$\hat{H} = \sum_i^N \frac{1}{2}\left(-\widetilde{\nabla}_i^2 + \tilde{x}_i^2 + \tilde{y}_i^2 + \gamma^2\tilde{z}_i^2\right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{107}$$

with

$$\gamma = \frac{\omega_z}{\omega_{ho}} \tag{108}$$

Again the repulsive potential V_{int} stays unaffected by this operation since it either evaluates to 0 or ∞.

References

[1] J Dubois and H R. Glyde. Bose-einstein condensation in trapped bosons: A variational monte carlo analysis. *Physical Review A*, 63, 08 2000.

[2] Morten Hjorth-Jensen. Lecture notes computational physics ii: Quantum mechanical systems, 2019.

[3] Jon Nilsen, Jordi Mur-Petit, M Guilleumas, M Hjorth-Jensen, and Artur Polls. Vortices in atomic bose-einstein condensates in the large gas parameter region. *Physical Review A*, 71:053610, 05 2005.

[4] Walter Kauzmann. *Quantum chemistry: an introduction*. Elsevier, 2013.

[5] G.H. Givens and J.A. Hoeting. *Computational statistics*. Wiley series in probability and statistics. Wiley-Interscience, 2005.

[6] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[7] Stephen Marsland. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2011.

[8] Henrik Flyvbjerg and H.G. Petersen. Error estimates on averages of correlated data. *The Journal of Chemical Physics*, 91, 07 1989.

[9] Marius Jonsson. Standard error estimation by an automated blocking method. *Physical Review E*, 98, 10 2018.

[10] Jørgen Høgberget. Quantum monte-carlo studies of generalized many-body systems. Master's thesis, University of Oslo, 2013.